

Delay Aware Survivable Routing with Network Coding in Software Defined Networks

Alija Pašić*, Péter Babarcsi*

*MTA-BME Future Internet Research Group, High-Speed Networks Laboratory (HSNLab),
Budapest University of Technology and Economics (BME), {pasic, babarcsi}@tmit.bme.hu

Abstract—It was demonstrated in transport networks that network (diversity) coding can provide sufficient redundancy to ensure instantaneous single link failure recovery, while near-optimal bandwidth efficiency can be reached. However, in the resulting multi-path routing problem the end-to-end delays were not considered. On the other hand, even in a European-scale network the delay difference of the paths has severe effect on the Quality-of-Service of application scenarios, such as video streaming. Thus, in this paper we thoroughly investigate survivable routing in Software Defined Networks (SDNs) with several additional delay bounds to the bandwidth cost minimization problem. We build on the fact that, if the user data is split into at most two parts, then the minimum cost coding solution has a well-defined acyclic structure of subsequent paths and disjoint path-pairs between the communication end-points. Complexity analysis and integer linear programs are provided to solve these delay aware survivable routing problems in SDNs.

Index Terms—delay aware routing, survivable routing, network coding, instantaneous recovery, transport networks

I. INTRODUCTION

Lately, delay is getting more and more into the spotlight in transport networks, because of the requirements of new applications (e.g., telesurgery, stock market, VoIP, etc.) which are highly delay sensitive besides they require high resilience. Satisfying both constraints at the same time in a bandwidth-efficient way is unquestionably one of the most challenging tasks of service providers in legacy networks. Luckily, the separation of the control plane from the data plane in Software Defined Networks (SDN) eases the control of the user flows, and enables network operators to steer the traffic to middleboxes performing specialized tasks. Furthermore, by using Virtual Network Functions (VNFs), these middleboxes can be invoked on demand in a virtualized environment, without the long lasting deployment of specialized hardware components at specific nodes.

The freedom provided in an SDN/VNF environment enables the proliferation of more complicated network operations. Multi-path routing is thoroughly investigated at several layers of the TCP/IP protocol stack [1], [2], and could be a real choice for service providers in the Quality-of-Service (QoS) support by offering sufficient path diversity for resilience, traffic engineering, higher throughput, etc. In a multi-path Internet, survivability is ensured in a fairly natural way (i.e., through disjoint end-to-end paths), and the user might choose a path with the lowest possible delay. However, in order to use network resources (e.g., bandwidth) in an efficient manner, further techniques (middleboxes/VNFs) should be

deployed in the network performing, e.g., *network coding*. One such survivable network coding deployment was shown in [3], where the necessary VNFs for network coding were installed in the SDN facility of the pan-European research and education network (GÉANT).

In [3] the high QoS requirement was defined as *instantaneous recovery*, i.e., after-failure signaling is completely eliminated from the recovery process. In other words, no flow rerouting or packet retransmission is required upon a single link failure (which are most common in transport networks). Optimal bandwidth efficiency of dedicated protection approaches with instantaneous recovery was investigated, and shown that in order to reach that the user data might be split into arbitrary many parts. Although suitable for a theoretical lower bound, from a practical point of view (e.g., network equipment and management complexity) this can not be implemented. Hence, survivable routing with diversity coding (SRDC) was introduced in [4] in which the *user data is divided into at most two parts* in order to ensure instantaneous recovery, while approaching the theoretical lower bound in bandwidth efficiency. In [5] it has been proven that *every minimum cost SRDC solution can be decomposed into three end-to-end directed acyclic graphs (DAGs)*, forwarding the two data parts (A and B) along some redundancy data ($A \oplus B$, i.e., the eXclusive OR), respectively.

Combining the results of [4], [5], our survivable routing problem turns into finding three appropriate DAGs between the communication endpoints s and t . With SRDC instantaneous recovery is ensured as the data transmitted on the DAGs is unchanged upon an arbitrary failure occurs. In [6] the above benefits of SRDC were demonstrated through a video streaming application scenario in GÉANT, but it was also noted that the end-to-end delay and the delay difference between the DAGs has severe effect on the performance of video streaming. Thus, considering the delay difference of the DAGs in the optimization problem is the last step to make SRDC work in a wide range of SDN transport networks, and will be made in this paper. We define the delay of a single end-to-end DAG in our SDN implementation, and investigate several additional delay constraints to the bandwidth cost minimization problem corresponding to different QoS scenarios.

The rest of the paper is organized as follows. In Section II the related work is presented focusing on QoS routing and differential delay aware routing. In Section III the preliminaries and problem formulation for survivable routing is discussed

in details. Section IV introduces our integer linear programming solution to the delay aware routing problem, while Section V presents complexity results when capacity constraints are present in the network. Experimental results are shown in Section VI, and the paper is concluded in Section VII.

II. RELATED WORK

A. QoS Routing

Seeking for a minimum cost (or shortest) path while satisfying an additional constraint (e.g., delay or jitter) is a fundamental problem and arises in several applications, referred to as Quality-of-Service (QoS) routing [7]. Note that, finding a single path or a pair of disjoint paths while minimizing a single objective (e.g., cost or length) can be solved in polynomial time with Dijkstra's and Suurballe's [8] algorithm, respectively. On the other hand, finding a single path satisfying multiple constraints is already an NP-hard problem [9], called the shortest weight-constrained path problem, where a minimum cost path is required between the source s and destination t , such that the delay of the path is lower than a pre-defined bound. Algorithms for this problem, i.e., finding minimum cost path that satisfies a certain (delay) constraint were presented in [10].

In order to extend this work to survivable routing, Orda and Sprintson investigated the problem of finding a constrained shortest link-disjoint path pair [11], and they proved that this problem is also NP-hard. In their 2-Restricted Link Disjoint Paths (2DP) problem they minimized the total cost of the paths while both paths have to obey a specific delay bound D . Several approximations were presented in [11] for the 2DP problem. To further generalize, Xiao et. al. introduced the problem of finding a set of k link-disjoint paths from s to t [12], such that the total cost of these paths is a minimum and that the delay for each path is not greater than a specified bound (T). Of course this problem contains the problem of 2DP, thus, it is also NP-hard [11]. Besides obeying a delay bound for each individual path, in [12] a more general network programming based approach was presented for finding k constrained shortest link-disjoint paths, such that the *overall delay of these paths* should be lower than a specified bound (kT). Algorithms were proposed to solve the relaxed versions of these problems, and the equivalency of the two relaxed problems was shown as well.

B. Differential Delay Aware Routing

Although the delays of individual paths in QoS routing is an important question, from a practical point of view the difference between the path delays could be a more serious issue in some application environments. For example, with the deployment of next-generation SONET/SDH technology virtual concatenation (VC) enabled service providers to split the traffic of a single circuit into multiple finer granularity parts, and route these parts along multiple paths. However, besides of the several advantages the application of VC provides, it introduces *differential delay (DD)* among the diversely routed paths as well, which boiled down to the issue of increased

buffer size at the destination node for DD compensation. In order to avoid service degradation, differential delay of the paths should be considered in the routing problem, as in optical networks the maximal DD compensation is about 125 ms with off-chip SDRAM technology [13].

The authors in [14] introduced the Two-Sided Constrained Path (TSCP) problem, where the task is to decide whether a new VC can be added to a VC group. Formulating with the DD constraint, the task is finding a path with overall delay of D between a given minimum and a maximum bound, i.e., $D_{min} < D < D_{max}$. It was proved that the TSCP problem is NP-hard [14]. In [15], the DD is defined as the difference between the delay of the highest and smallest delay paths. In their Differential Delay Routing (DDR) problem the task is to find a given number of paths, while their DD is lower than a pre-defined delay bound. It was shown that minimizing the delay difference of paths in DDR is not only NP-hard but provably hard to approximate within a constant factor. In [16] the same authors introduce cumulative differential delay, which is the sum of the differences of delays of all the paths of a solution compared to the highest delay path.

In the previous works the objective function was to minimize the differential delay, while neither link costs nor the disjointness of the paths were considered in the optimization. The study in [13] extended DD aware multi-path routing with survivability. Contrary to the differential delay problems above, their goal is to minimize the total cost of the paths while the disjointness of these paths is required in order to ensure single link failure resilience. The mathematical formulation of the survivable multi-path DD constrained routing problem has been presented in [13], where a DD bound have to be satisfied between each pair of the k paths. The NP-completeness of this problem follows from the DDR problem [15].

III. PRELIMINARIES AND PROBLEM FORMULATION

In our *Delay Aware Routing with Network Coding (DARC)* problem the network is represented by a directed graph $G = (V, E, k, c, d)$ with node set V , link (arc) set E , and three additional attributes for each link $e \in E$: the capacity $k(e) \in \mathbb{N}$, i.e., the number of bandwidth units available for data transmission; a non-negative cost function $c(e) \in \mathbb{R}^+$, which is defined as the cost of using one unit of bandwidth along link e ; and the delay $d(e) \in \mathbb{N}$, which corresponds to the time transmitting data between the end nodes of the link. As part of the input of the delay aware survivable routing problem a connection request $C = (s, t, b, D)$ is given, which consists of the source node $s \in V$, destination node $t \in V$, the number of bandwidth units b requested for data transmission, and the delay bound D . Before we formulate our DARC problem, we recall here the main findings of SRDC [4], which gives the starting point of our work. Throughout this paper, we build on the following definition of survivable routing [4]:

Definition 1. We say that $R = (V^R, E^R, f)$ is a *survivable routing* of a connection $C = (s, t, b, D)$ in G with flow values f (where $V^R \subseteq V$, $E^R \subseteq E$, and $\forall e \in E^R : f(e) \leq k(e)$), if

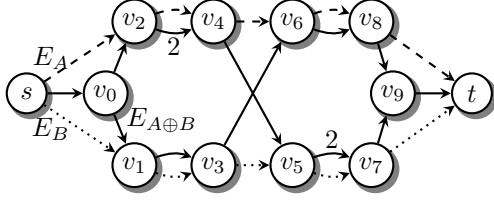


Fig. 1. A survivable SRDC solution for request $C = (s, t, 2, -)$ with the corresponding routing DAGs E_A , E_B and $E_{A \oplus B}$. Links with $f(e) = 2$ are duplicated. Link costs are unit. The link delays are $d(e) = 1$, or written next to the arc otherwise.

there is an $s - t$ flow of value $F \geq b$ in R , even if we delete any single link of R .

Our objective is to minimize the total bandwidth cost of the solution, formally:

$$\min_R \sum_{e \in E^R} c(e) \cdot f(e). \quad (1)$$

A minimum cost SRDC solution for the input $C = (s, t, b = 2, -)$ with respect to Eq. (1) has $\forall e \in E^R : f(e) \leq 2$ (consequence of [5, Theorem 1])¹. Furthermore, E^R always can be decomposed into three link sets $E_A, E_B, E_{A \oplus B}$, respectively transmitting data parts A and B and $A \oplus B$, called **routing DAGs**. Thus, if we delete an arbitrary link from an SRDC solution $e \in E^R$, there remains at least two routing DAGs in which a path connects s to t (satisfying [5, Theorem 2]) in order to guarantee instantaneous recovery (i.e., no flow rerouting is required on the intact links). An example is presented in Fig. 1. If a link with $f(e) = 1$ fails, e.g., (s, v_1) , it disrupts only a single routing DAG (E_B) and the above claim trivially holds. Further note, that the failure of a link $f(e) = 2$, e.g., (v_1, v_3) although affects two routing DAGs, only E_B fails, while on $E_{A \oplus B}$ the path $s \rightarrow v_0 \rightarrow v_2 \rightarrow v_4 \rightarrow v_5 \rightarrow v_7 \rightarrow v_9 \rightarrow t$ still ensures $s - t$ connectivity without any reconfiguration of the network; however, possibly with an increased delay (from 7 to 9 units in Fig. 1).

Besides instantaneous recovery, SRDC provides additional benefits to the connections in the failure-less state as well. In multi-path routing (or shared path protection approaches [13]) we have to wait for the highest delay path to reconstruct user data. On the other hand, the additional redundancy in SRDC provided by network coding ensures that the two lower delay paths determine the delay of the connection (as we can reconstruct user data from arbitrary two of the three routing DAGs).

A. Delay of a Routing DAG in SDNs

It has been proven in [4] that each routing DAG in a minimum cost SRDC solution consists of series of paths (\mathcal{P}) and disjoint path-pairs, called **islands** (\mathcal{I}), and that each island is at most part of one routing DAG. In Figure 1 E_A and E_B

have $\mathcal{P}_{E_A} = \mathcal{P}_{E_B} = \{s \rightarrow t\}$ path. On the other hand, $E_{A \oplus B}$ consist of $\mathcal{P}_{E_{A \oplus B}} = \{s \rightarrow v_0, v_9 \rightarrow t\}$ paths, and $\mathcal{I}_{E_{A \oplus B}} = \{v_0 \rightarrow v_9\}$ disjoint path-pair (island with “splitter node” v_0 and “merger node” v_9).

The SDN implementation in [3] was built on this important structural property of the optimal routing DAGs. Note that the implementation of the merger VNF (i.e., which selects among the two copies of the same data part) highly affects the overall delay of the routing DAG in the failure-less state and after a single link failure occurs, too. In order to eliminate signaling from the recovery process (i.e., ensure instantaneous recovery) a link failure should be oblivious to the merger node, i.e., it has to switch from the failed path autonomously to the operating path. The merger VNF in [3] keeps track of the highest sequence number of the forwarded packets (SEQ_{FW}). A packet p is only forwarded on the merger’s outgoing link if its sequence number SEQ_p is larger than SEQ_{FW} and set $\text{SEQ}_{FW} = \text{SEQ}_p$. As a result, a merger forwards the packets from the “faster” path from the two disjoint paths of an island (I_{\min}) in a failure-less state, and discards the duplicates that arrive on the “slower” path of the island (I_{\max}). On the other hand, if a failure occurs on the faster path, the merger will forward the packets on the slower path automatically.

In order to capture the delay characteristics of the routing DAGs, we introduce two delay values for each island I : $d_{\min}^I = \sum_{e \in I_{\min}} d(e)$ corresponding to the delay of the island in the failure less state (i.e., the faster path); and the delay difference between the two disjoint paths $\Delta^I = \sum_{e \in I_{\max}} d(e) - \sum_{e \in I_{\min}} d(e)$ corresponding to the delay increase upon a failure occurs on the faster path. Thus, the end-to-end delay of a routing DAG can be modeled as

$$\delta_{E_j} = \sum_{P \in \mathcal{P}_{E_j}} \sum_{e \in P} d(e) + \sum_{I \in \mathcal{I}_{E_j}} d_{\min}^I \quad (2)$$

in the failure-less state, while it increases to

$$\Delta_{E_j} = \delta_{E_j} + \max_{I \in \mathcal{I}_{E_j}} \Delta^I. \quad (3)$$

in worst case upon a failure along the island with the largest delay difference between its two paths. In Figure 1, $\delta_{E_{A \oplus B}} = 7$. As $\Delta^I = 2$ for the single island $v_0 \rightarrow v_9$, $\Delta_{E_{A \oplus B}} = 9$.

B. Investigated Delay Bounds

In order to cover most delay scenarios, we define four different delay constraints to extend SRDC and define our delay-aware routing problem DARC in SDNs. Formally, DARC minimizes the total cost of a survivable routing Eq. (1), while the following additional delay bounds must be satisfied by the routing DAGs.

- (i) QoS routing: the after-failure delay for each routing DAG is less than a given bound D_p [11]:

$$\forall j \in \{A, B, A \oplus B\} : \Delta_{E_j} \leq D_p.$$

¹Remember that in SRDC we divide user data at most into two parts A and B without delay constraints.

TABLE I

THE END-TO-END DELAY DIFFERENCE ON THE TWO FASTEST ROUTING DAGS UPON A SINGLE LINK FAILURE (WLOG $\delta_{E_A} \leq \delta_{E_B} \leq \delta_{E_{A \oplus B}}$).

disrupted \ inc. delay	\emptyset	E_A	E_B	$E_{A \oplus B}$
\emptyset	$ \delta_{E_A} - \delta_{E_B} $	$ \delta_{E_B} - \min\{\Delta_{E_A}, \delta_{E_{A \oplus B}}\} $	$ \delta_{E_A} - \min\{\Delta_{E_B}, \delta_{E_{A \oplus B}}\} $	$ \delta_{E_A} - \delta_{E_B} $
E_A	$ \delta_{E_B} - \delta_{E_{A \oplus B}} $	—	$ \Delta_{E_B} - \delta_{E_{A \oplus B}} $	$ \delta_{E_B} - \Delta_{E_{A \oplus B}} $
E_B	$ \delta_{E_A} - \delta_{E_{A \oplus B}} $	$ \Delta_{E_A} - \delta_{E_{A \oplus B}} $	—	$ \delta_{E_A} - \Delta_{E_{A \oplus B}} $
$E_{A \oplus B}$	$ \delta_{E_A} - \delta_{E_B} $	$ \Delta_{E_A} - \delta_{E_B} $	$ \delta_{E_A} - \Delta_{E_B} $	—

- (ii) QoS routing: the overall after-failure delay of the three routing DAGs is less than a given bound D_o [12]:

$$\Delta_{E_A} + \Delta_{E_B} + \Delta_{E_{A \oplus B}} \leq D_o.$$

- (iii) DD aware routing: the delay difference between the two lower delay routing DAGs is under a specific bound D_w , corresponding to the buffer size in the operational state of the network [13] (wlog $\delta_{E_A} \leq \delta_{E_B} \leq \delta_{E_{A \oplus B}}$):

$$|\delta_{E_A} - \delta_{E_B}| \leq D_w.$$

- (iv) DD aware routing: the delay difference between the two fastest routing DAGs is under a specific bound $\leq D_f$ [13] corresponding to the maximal buffer size when an arbitrary single link failure occurs. As shown in Fig. 1, a single link failure can cause the disruption of a routing DAG, increase its end-to-end delay or both at the same time. All possible delay differences have to be considered, introduced in Table I.

IV. DELAY AWARE ROUTING WITH NETWORK CODING WITH INFINITE CAPACITIES

Remember that in a minimum cost SRDC solution $\forall e \in E^R : f(e) \leq 2$. Thus, in practice infinite link capacities mean $\forall e \in E : k(e) = 2$. Based on the delay characteristics of the routing DAGs, in Section IV-A we present our equivalent graph transformation of DARC to the disjoint-paths problem. In Section IV-B we present an Integer Linear Program (ILP) to solve the delay aware survivable routing problem on the transformed graph.

A. Equivalent Graph Transformation for DARC

After the graph transformation in [4] the minimum cost survivable routing problem of SRDC-I is traced back to finding three link-disjoint $s-t$ paths in an auxiliary graph G^* . Although this makes SRDC-I polynomial-time solvable, the graph transformation *cannot handle the delay bounds D on the routing DAGs of DARC*. Thus, we need to extend it as follows.

The input of DARC is the graph $G = (V, E, k, c, d)$ with delay values $d(e)$ for every link e . An auxiliary (multi-)graph $G^* = (V, E^*, c^*, d_{min}, \Delta)$ is created, where:

- Node set V is the same as in G .
- Links E^* are the links of G . Additional *virtual links* $e_{(u,v)}$ representing potential islands I are added between every pair of distinct node-pairs, i.e., a potential island with splitter node u and merger node v .

- The cost of $c^*(e_{(u,v)})$ is set to the cost of a minimum cost disjoint path-pair between nodes u and v in G (calculated with Suurballe's algorithm). The original links of G have the same cost ($\forall e \in E : c^*(e) = c(e)$).
- We have to capture the routing DAG delays in Eq (2)-(3) in our graph transformation. Thus, two variables d_{min}^I and Δ^I are introduced for each virtual link (island) $I = e_{(u,v)}$ (i.e., the delay of I_{min} and the delay difference between I_{min} and I_{max}). For links $e \in E$ we define $d_{min}^e = d(e)$, and $\Delta^e = 0$.

For example, 10 virtual links are added in Fig. 1, e.g., the virtual link $e = (v_0, v_9)$ representing potential island with splitter v_0 and merger v_9 has $c^*(e) = 12$, $d_{min} = 5$, $\Delta = 2$. A DARC solution in G^* can be easily transformed back to G by replacing the virtual links $e_{(u,v)}$ with the corresponding islands. Thus, DARC is now reduced to finding three link-disjoint $s-t$ paths in G^* that fulfill the additional delay bounds formulated in Problems (i)-(iv). Although NP-completeness of finding three link-disjoint paths with additional delay bounds immediately follows from previous works [9], [11], [13], [15], owing to the correlation between the links and link parameters of G^* they are not directly applicable to our problem. However, our conjecture is that DARC inherits the NP-completeness of these problems. Thus, in Section IV-B we present ILPs for solving DARC for all of these delay bounds.

B. Integer Linear Program for DARC

Here, we present a general ILP formulation for DARC based on [4], [13] for finding three minimum cost link-disjoint paths in $G^* = (V, E^*, c^*, d_{min}, \Delta)$ with additional delay bounds required for QoS routing and for DD aware routing. The connection request is $C = (s, t, b = 2, D)$. The *three paths corresponding to the routing DAGs* are denoted as $w \in \{A, B, A \oplus B\} = \mathcal{W}$, respectively. Binary variables $x^w(e)$ are used to indicate the paths for each routing DAG. Our objective is to minimize the total bandwidth cost in terms of Eq. (1):

$$\min \sum_{w \in \mathcal{W}} \sum_{e \in E} c^*(e) \cdot x^w(e).$$

The following constraints are required to find a survivable routing:

$$\forall w \in \mathcal{W}, \forall i \in V:$$

$$\sum_{(i,j) \in E} x^w(i,j) - \sum_{(j,i) \in E} x^w(j,i) = \begin{cases} 1 & , \text{ if } i = s \\ -1 & , \text{ if } i = t \\ 0 & , \text{ otherwise} \end{cases}, \quad (4)$$

$$\forall e \in E: \sum_{w \in \mathcal{W}} x^w(e) \leq 1, \quad (5)$$

$$\forall w \in \mathcal{W}, \forall e \in E: x^w(e) \cdot \Delta^e \leq y^w. \quad (6)$$

Constraint (4) formulates the flow conservation for each path w . Constraint (5) ensures that the disjointness of the paths. Constraint (6) gives a lower bound for the integer variable y^w , which captures the worst case delay increase of path (routing DAG) w upon a single link failure formulated in Eq. (3). For the different delay requirements, we have to add the following constraints:

- Problem (i): Our bound keeps the after-failure delay of the individual routing DAGs on a tolerable level even for the largest delay increase.

$$\forall w \in \mathcal{W}: \sum_{e \in E} x^w(e) \cdot d_{min}^e + y^w \leq D_p. \quad (7)$$

- Problem (ii): We keep the overall after-failure delay in each failure scenario under a specific bound.

$$\sum_{w \in \mathcal{W}} [y^w + \sum_{e \in E} x^w(e) \cdot d_{min}^e] \leq D_o. \quad (8)$$

- Problem (iii): To capture the delay difference between the two fastest (i.e., lowest delay) routing DAGs, we formulate the order of the DAGs ($wlog \delta_{E_A} \leq \delta_{E_B} \leq \delta_{E_{A \oplus B}}$) in Constraints (9)-(10). Thus, the delay bound corresponds to paths x^A and x^B in Constraint (11).

$$\sum_{e \in E} x^A(e) \cdot d_{min}^e \leq \sum_{e \in E} x^B(e) \cdot d_{min}^e, \quad (9)$$

$$\sum_{e \in E} x^B(e) \cdot d_{min}^e \leq \sum_{e \in E} x^{A \oplus B}(e) \cdot d_{min}^e, \quad (10)$$

$$\sum_{e \in E} [x^B(e) - x^A(e)] \cdot d_{min}^e \leq D_w. \quad (11)$$

- Problem (iv): In addition to Constraints (9)-(11), in order to formulate all possible delay differences between the two fastest paths upon a single link failure occurs, we have to formulate all possible situations in Table I. For example, Constraints (12)-(13) formulate when E_B is disrupted while the end-to-end delay of E_A is increased (as we don't know which one has lower delay):

$$y^A + \sum_{e \in E} [x^A(e) - x^{A \oplus B}(e)] \cdot d_{min}^e \leq D_f, \quad (12)$$

$$-y^A + \sum_{e \in E} [x^{A \oplus B}(e) - x^A(e)] \cdot d_{min}^e \leq D_f. \quad (13)$$

V. DELAY AWARE SURVIVABLE ROUTING WITH CAPACITY CONSTRAINTS

The SRDC problem can be traced back to the 3-disjoint paths problem resulting a polynomial-time complexity if all the links can support at least two routing DAGs [4] ($\forall e \in E : k(e) \geq 2$). However, the complexity of the capacity constrained case (i.e., there are some *bottleneck links* with sufficient capacity only for a single routing DAG $\exists e \in E :$

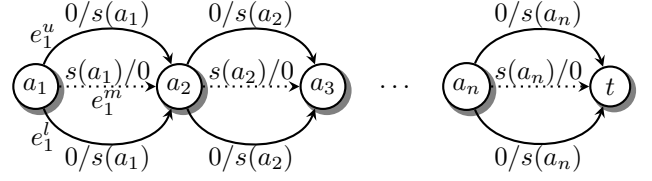


Fig. 2. Transformation of Partition (and Bin Packing) to DARC. Link delays/link costs are shown next to the links. Note that the bold links have $k(e) \geq 2$, while the dotted links have $k(e) = 1$.

$k(e) = 1$) is an open problem². For DARC with capacity constraints, with the selection of $\forall e \in E : k(e) = 1$, the problem is to find three disjoint paths with specific delay bounds, as the routing DAGs are simple $s \rightarrow t$ paths. The NP-completeness proof of [15] for link-disjoint paths in DAGs can be easily transformed for three paths, proving the NP-completeness of Problems (i)-(iv). In the rest of the section, we show an alternative transformation for routing DAGs when not all links are bottlenecks for Problems (i)-(ii).

Theorem 1. *To decide whether a $\leq Z$ cost DARC solution exists with bottleneck links is NP-complete for Problem (ii).*

Proof: DARC is in NP, a solution with $\leq Z$ cost with $\leq D_o$ is a proof.

Assuming we are given an instance of the Partition Problem [9], that is, a finite set A of items with size $s(a) \in \mathbb{Z}^+$ for each $a \in A$. Let us denote $T = \sum_{a \in A} s(a)$. Is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$? The polynomial time transformation for Partition with $|A| = n$ to DARC is given as follows (shown in Figure 2). We construct a graph with $n+1$ nodes using the following gadget for each a_i : we add three links (upper, middle and lower) $e_i^u = e_i^m = e_i^l = (a_i, a_{i+1})$, $i = 1, 2, \dots, n$ (with $t = a_{n+1}$). We define link delay and link cost values according to $s(a_i)$, as shown in Figure 2. The dotted links (e_i^m) are bottleneck links, and we define $Z = 3T$ and $D_o = T/2$ for the connection request $C = (s = a_1, t, 2, D_o)$. As we are proving overall delay, we won't distinguish between the paths \mathcal{P} and islands \mathcal{I} of different routing DAGs, unless other specified.

(\Rightarrow) We show how to convert a DARC solution to Partition. For every gadget $f(e_i^u) \geq 1$ and $f(e_i^l) \geq 1$ follows, as single link failure survivability cannot be guaranteed otherwise, resulting $\sum_{a_p \in \mathcal{P}=A} 2s(a_p) = 2T$. In order to satisfy the overall delay bound $T/2$, islands \mathcal{I} are created (with $\Delta^I = 0$), which do not use e_i^m on gadgets $\sum_{a_i \in \mathcal{I}} s(a_i) \geq T/2$. Although creating an island in gadget a_i reduces overall delay from $s(a_i)$ to 0, but it doubles the total cost from $2s(a_i)$ to $4s(a_i)$ at the same time. Thus, the cost increase owing to the islands is $\sum_{a_i \in \mathcal{I}} 2s(a_i) \geq T$. Hence, the total cost is $\sum_{a_i \in \mathcal{I}} 2s(a_i) + \sum_{a_p \in \mathcal{P}=A} 2s(a_p) \geq 3T$, which means if a solution exist to the DARC problem with $\leq 3T$ cost, it must be a minimum cost solution with $\sum_{a_i \in \mathcal{I}} 2s(a_i) = T$.

²Note that, the restricted version of the capacity constrained problem, where additional nodal constraints are in place [4], is NP-complete.

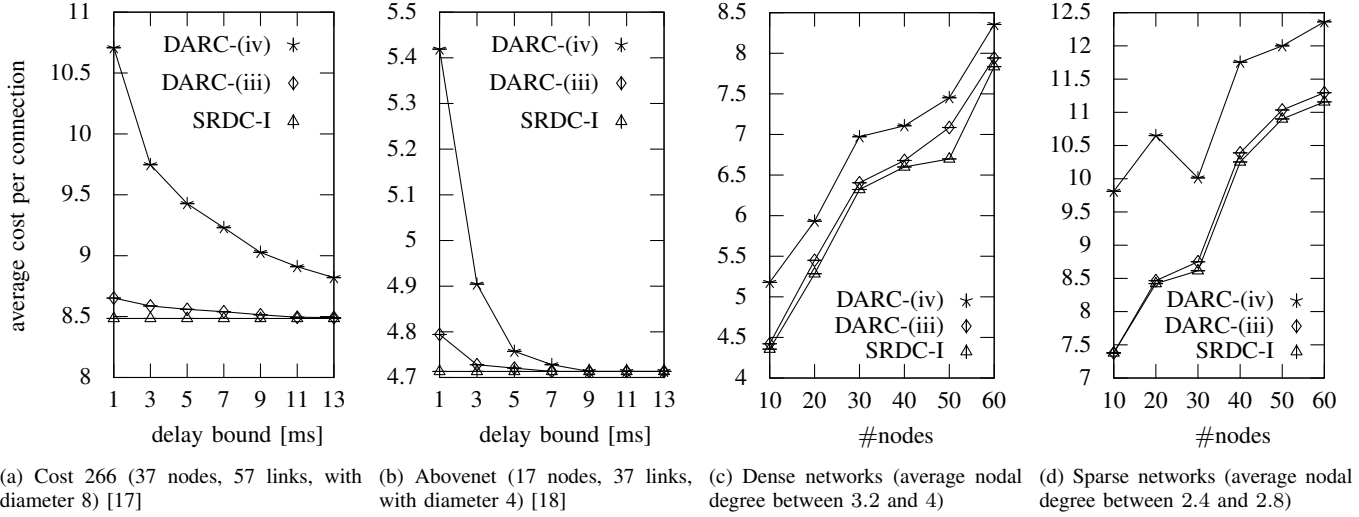


Fig. 3. Bandwidth cost of the DD problems as a function of the delay bound in real-world and real-world like topologies

As a corollary, $f(e_i^m) = 0$ indicates island creation for $\sum_{a_i \in \mathcal{I}} s(a_i) = T/2$ in a minimum cost DARC solution, i.e., the value of $f(e_i^m)$ defines partitions A' and $A \setminus A'$.

(\Leftarrow) In the other direction, it is easy to convert a Partition to three routing DAGs. Let assume wlog, that $\mathcal{P}_{E_A} = \{a_0 \rightarrow a_{n+1}\}$ using $\forall a_i \in A : e_i^u$, $\mathcal{P}_{E_B} = \{a_0 \rightarrow a_{n+1}\}$ using $\forall a_i \in A : e_i^l$. We define $\mathcal{P}_{E_{A \oplus B}} = \{a_i \rightarrow a_{i+1}\}$ with $\forall a_i \in A' : f(e_i^m) = 1$, and $\mathcal{I}_{E_{A \oplus B}} = \{a_j \rightarrow a_{j+1}\}$ with $\forall a_j \in A \setminus A' : f(e_j^m) = 0$. Thus, Partition gives a DARC a solution, which finishes the proof. ■

Theorem 2. *To decide whether a $\leq Z$ cost DARC solution exists with bottleneck links is NP-complete for Problem (i).*

As the reasoning is very similar to Theorem 1, we give only the idea of the proof here. Assuming we are given an instance of the Bin Packing Problem [9], that is, a finite set A of items with size $s(a) \in \mathbb{Z}^+$ for each $a \in A$, a positive bin capacity D and a positive integer K . Is there a partition of A into A_1, A_2, \dots, A_K such that the sizes of the items in each A_i is D or less. The problem remains NP-complete with fixed K , thus, we set $K = 3$. The polynomial time transformation is the same as for Theorem 1, with the difference that we define $Z = 2T$ and $D_p = D$. Now, owing to the survivability requirement $\sum_{a_p \in \mathcal{P}=A} 2s(a_p) = 2T$. Thus, a $\leq Z$ solution has minimum total cost, and each gadget is traversed by $f(e_i^u) = f(e_i^m) = f(e_i^l) = 1$, making $\delta_{E_j} = \Delta_{E_j}$ for each routing DAG. As only e_i^m has non-zero delay, the only question is which routing DAG suffers the delay $s(a_i)$ on link e_i^m . If we can keep $\delta_{E_j} \leq D_p$ for each routing DAG, then the e_i^m links traversed by routing DAGs E_A, E_B and $E_{A \oplus B}$ corresponds to the elements in the three bins, respectively, and vice versa.

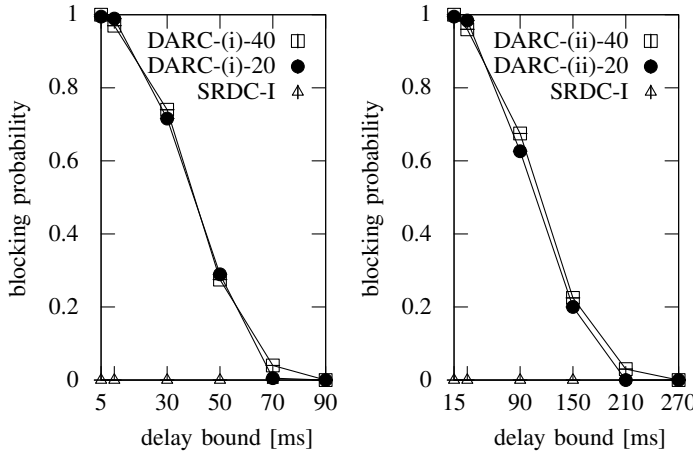
VI. EXPERIMENTAL RESULTS

In this section we investigate the total bandwidth cost of DARC, a practically useable survivable routing, i.e., when not only bandwidth cost is considered but also additional delay constraints are introduced for every connection request. We

compare our methods with SRDC, i.e., with the method where only the bandwidth cost is minimized and no additional constraints are given. Of course the cost increases by introducing additional bounds, but in exchange for that we can guarantee a certain level of QoS, not only in terms of reliability but also in terms of end-to-end delay. This could dramatically improve the user experience of video streaming in SDNs, which leads in long term to higher revenue and competitive advantage (compared to other providers).

We investigated random generated real-like planar $G = (V, E, k, c, d)$ network topologies with different sizes and densities, and some real world topologies, too. All of the arcs have unit cost ($\forall e \in E : c(e) = 1$). Furthermore, the delay of the arcs $d(e)$ is a function of the distance between its adjacent nodes, and scaled into range of 1 and 25 ms. These values are based on the measurements taken in optical transport networks and in the SDN facility of GÉANT [3], [13]. The arc capacities were set high enough so that no blocking occurs due the capacity deficit (i.e., lack of resources). 200 connection requests $C = (s, t, 2, D)$ were generated randomly with a given delay bound. Note that, the limited request number is a consequence of the high computational complexity of our ILPs. With 200 demands we enabled our ILP to run for middle-scale topologies in a reasonable time.

In Figure 3 the simulation results of the DD Problems (iii)-(iv) are presented, denoted as DARC-(iii) and DARC-(iv), respectively. Fig. 3a-3b present the total bandwidth cost of real-world topologies depending on the delay bound. It can be observed that as the delay bound decreases the total cost of DARC-(iv) increases dramatically. It is foreseeable because DARC-(iv) takes all possible failure scenarios into account. This means that if any single link failure event occurs, this method provides a solution within the given differential delay bound. In the end, it sacrifices the cost efficiency in order to find three routing DAGs approximately with the same delay. For DARC-(iii) only the two lower delay routing DAG-s are



(a) Blocking probability of DARC-(i) (b) Blocking probability of DARC-(ii)

Fig. 4. Blocking probability of the QoS routing problems as a function of the delay bound in maximum planar graphs with node number 20 and 40

considered in the operational state. Thus, its cost increase is much smoother.

Fig. 3c-3d plot the total bandwidth cost in networks with different sizes and densities for a given delay bound (9 ms). The tendencies are the same as in real world topologies, i.e., DARC-(iv) needs much more resources to satisfy all of the constraints related to all single link failures, while DARC-(iii) does not require too much extra resources, independently from network size. Note that, the randomly generated traffic demands can cause some bandwidth cost fluctuation (as in larger networks not all $s - t$ pairs are considered as requests).

In Figure 3 for differential delay no blocking occurs, which can be explained with the fact that by optimizing for differential delay we have lot more options to explore. On the other hand, in QoS routing where a strict bound is given on path-length, and if there is no path satisfying this bound in the network, then the connection request has to be blocked. This leads us to the recognition that in the QoS routing problems the blocking probability characterizes the problem better than total cost. This value could be an indicator for the network operator, i.e., what percentage of the request can be satisfied with a given QoS level. According to that, in Fig. 4a-4b the blocking probabilities of DARC-(i) and DARC-(ii) are shown, respectively. One can observe that for both QoS cases the blocking probability increases rapidly after a given delay bound is reached. As mentioned before, this is due the fact that if there are no paths shorter than a given bound, then the request gets blocked.

VII. CONCLUSION

In the practical deployment of survivable routing with diversity coding (SRDC) in the SDN facility of GÉANT it was shown that the end-to-end delay difference of the routing DAGs has severe effect on the performance of video streaming [6]. Thus, in this paper we defined the delay of routing DAGs, and introduced Delay Aware Routing with

Network Coding (DARC) which captures several QoS routing and differential delay aware bounds of survivable routing in SDNs. We demonstrated that the problem can be transformed into finding three end-to-end paths with some additional requirements. An ILP formulation was presented to find the routing DAGs with four different set of delay constraints. We presented alternative complexity proofs for the capacity constrained case, when some but not all links are bottlenecks in the network. Through simulations on small- and medium-scale network topologies we demonstrated the effect of the different delay bounds on the total bandwidth cost of the optimal solution.

ACKNOWLEDGMENTS

Research of P. Babarczi was partially supported by the Hungarian Scientific Research Fund (OTKA grant K108947). P. Babarczi was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences (MTA).

REFERENCES

- [1] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath tcp," *SIGCOMM CCR*, vol. 41, no. 4, pp. 266–277, 2011.
- [2] B. Augustin, T. Friedman, and R. Teixeira, "Measuring multipath routing in the internet," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 830–840, 2011.
- [3] P. Babarczi, A. Pasic, J. Tapolcai, F. Németh, and B. Ladóczki, "Instantaneous recovery of unicast connections in transport networks: Routing versus coding," *Elsevier Computer Networks*, vol. 82, pp. 68–80, 2015.
- [4] A. Pasic, J. Tapolcai, P. Babarczi, E. Bérczi-Kovács, Z. Király, and L. Rónyai, "Survivable routing meets diversity coding," in *IFIP Networking*, 2015, pp. 1–9.
- [5] P. Babarczi, J. Tapolcai, L. Rónyai, and M. Médard, "Resilient flow decomposition of unicast connections with network coding," in *Proc. IEEE Intl. Symp. on Information Theory (ISIT)*, 2014, pp. 116–120.
- [6] B. Ladóczki, C. Fernandez, O. Moya, P. Babarczi, J. Tapolcai, and D. Guija, "Robust network coding in transport networks," in *34th IEEE INFOCOM Demo session*, 2015, pp. 1–2.
- [7] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.
- [8] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325–336, 1984.
- [9] M. R. Garey and D. S. Johnson, "Computers and intractability: a guide to np-completeness," 1979.
- [10] H. C. Joksche, "The shortest route problem with constraints," *Journal of Math. analysis and applications*, vol. 14, no. 2, pp. 191–197, 1966.
- [11] A. Orda and A. Sprintson, "Efficient algorithms for computing disjoint qos paths," in *23th IEEE INFOCOM*, vol. 1, 2004.
- [12] Y. Xiao, K. Thulasiraman, and G. Xue, "Constrained shortest link-disjoint paths selection: a network programming based approach," *IEEE Trans. on Circuits and Systems I*, vol. 53, no. 5, pp. 1174–1187, 2006.
- [13] S. Huang, C. U. Martel, and B. Mukherjee, "Survivable multipath provisioning with differential delay constraint in telecom mesh networks," *IEEE/ACM Trans. on Networking*, vol. 19, no. 3, pp. 657–669, 2011.
- [14] S. Ahuja, M. Krunk, and T. Korkmaz, "Optimal path selection for minimizing the differential delay in ethernet-over-sonet," *Computer Networks*, vol. 50, no. 13, pp. 2349–2363, 2006.
- [15] A. Srivastava, S. Acharya, M. Alicherry, B. Gupta, and P. Risbood, "Differential delay aware routing for ethernet over sonet/sdh," in *24th IEEE INFOCOM*, vol. 2, 2005, pp. 1117–1127.
- [16] A. Srivastava, "Flow aware differential delay routing for next-generation ethernet over sonet/sdh," in *IEEE ICC*, vol. 1, 2006, pp. 140–145.
- [17] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proc. INOC*, 2007.
- [18] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4, pp. 133–145, 2002.